A1

33. (AS NEW HEREIN) The multiprocessor system as claimed in claim 28, wherein:

the arbitrary system module generates the read request if a mishit occurs in the arbitrary system module;

the system module, other than the arbitrary system module, storing the data requested by the read request prereads the requested data: and

the system module, other than the arbitrary system module, transfers the preread data to the buffer within the crossbar module, with a priority lower than a priority of a normal data transfer between the system modules and the crossbar module.

34. (AS NEW HEREIN) The memory access method as claimed in claim 33, wherein the crossbar module transfers the preread data stored in the buffer thereof to the arbitrary system module with a priority lower than the priority of the normal data transfer between the system modules and the crossbar module.

REMARKS

In accordance with the foregoing, various of the pending claims 1-11 have been amended to improve form and without change of substance and, further, new claims 12-34 have been added, to afford a varying scope of protection for the invention.

No new matter is presented and, accordingly, approval and entry of the amended and new claims are respectfully requested.

STATUS OF CLAIMS

Claims 1-11 are pending in this Action and all thereof are rejected. The rejections are respectfully traversed.

ITEM 3: REJECTION OF CLAIMS 1 AND 5 UNDER 35 USC § 103(a) AS BEING UNPATENTABLE OVER CARPENTER ET AL. IN VIEW OF IRIE ET AL.

As noted by the examiner, Carpenter et al. does not show the use of a preread (or speculative read) and a buffer within the crossbar module to hold the data preread from a system module other than the arbitrary system module.

The examiner accordingly relies on Irie et al. as showing the use of a preread and a buffer within the crossbar module to hold the data preread from a system module other than the arbitrary system module. However, Irie et al. does not show the use of a buffer with a crossbar module to hold data preread from a system module other than an arbitrary system module.

As may be seen from the abstract of Irie et al., Irie et al. relates to a cache status report sum-up for a normal read in a multiprocessor system. Steps 905 and 908 shown in Fig. 9 and column 6, line 49 to column 7, line 12 and column 7, lines 44-50 referred to by the examiner are related to the operation of the coherency status report sum-up unit 50 within the SC board 30 shown in Fig. 1, and are not related to the operation of the crossbar unit 40. In addition, it is described that the coherency status report sum-up unit 50 makes a summary of the coherency status reports when all coherency status reports are received, and sends the coherency status summary CSS to the coherent read requesting processor board 10-0.

Moreover, Fig. 6 and the related description in lire et al. directed to the crossbar unit 40 within the SC board 30 do not show or mention a buffer to hold the data preread from a system module other than the arbitrary system module.

It is respectfully submitted that making a summary of the coherency status reports for the normal read is completely different from holding the data preread from a system module other than the arbitrary system module.

Moreover, it follows that there is no *prima facie* obviousness supporting the combination of the two references since Irie et al. does not even disclose the features which the Action concedes are absent in the disclosure of Carpenter et al. Therefore, it is submitted that claims 1 and 5 patentably distinguish over Carpenter et al. and Irie et al., taken singly or in any proper combination.

ITEM 4: REJECTION OF CLAIMS 2 AND 6 UNDER 35 USC § 103(a) AS BEING UNPATENTABLE OVER CARPENTER ET AL. IN VIEW OF IRIE ET AL. AND CHRISTIE

Claims 2 and 6 are respectively dependent upon the base claims 1 and 5 which patentably distinguish over Carpenter et al. and Irie et al., as discussed above.

Christie merely proposes a phase change monitor for monitoring processor resources to detect a phase change in the program being executed. The phase change monitor signals a prefetch unit to indicate the detected phase change, so that the prefetch unit selectively carries

out a prefetch in response to the detected phase change.

Christie does not teach or suggest the use of a preread and a buffer within the crossbar module to hold the data preread from a system module other than the arbitrary system module, as recited in the base claims 1 and 5.

The combination relied upon in item 3, for which p*rima facie* obviousness was lacking is now supplemented by Christie, which likewise is lacking *prima facie* obviousness.

Therefore, it is respectfully submitted that claims 2 and 6 patentably distinguish over Carpenter et al., Irie et al. and Christie, taken singly or in any proper combination.

ITEM 5: REJECTION OF CLAIMS 4 AND 8 UNDER 35 USC § 103(a) AS BEING UNPATENTABLE OVER CARPENTER ET AL. IN VIEW OF IRIE ET AL. AND HOOKS ET AL.

Claims 4 and 8 are respectively dependent upon the base claims 1 and 5, which patentably distinguish over Carpenter of al. and Irie et al. as discussed above.

Hooks et al. merely proposes a bus arbitration system having first and second bus masters and a bus arbiter. The first bus master is adapted to perform speculative pre-fetching, and has a first REQ signal for requesting ownership of the bus and an SP signal for indicating when the bus ownership request is for a speculative pre-fetch. The second bus master has a second REQ signal for requesting ownership of the bus. The bus arbiter assigns a higher priority to the second bus master in response to the SP signal when the first bus master asserts the first REQ signal and the SP signal and the second bus master asserts the second REQ signal.

Hooks et al. does not teach or suggest the use of a preread and a buffer within the crossbar module to hold the data preread from a system module other than the arbitrary system module, as recited in the base claims 1 and 5.

The Action furthermore fails to show *prima facie* obviousness of the combination relied upon.

Therefore, it is submitted that claims 4 and 8 patentably distinguish over Carpenter et al., Irie et al. and Hooks et al.

ITEM 6: REJECTION OF CLAIMS 3 AND 7 UNDER 35 USC § 103(a) AS BEING UNPATENTABLE OVER CARPENTER ET AL. IN VIEW OF IRIE ET AL, CHRISTIE AND HOOKS ET AL.

Claims 3 and 7 are respectively dependent upon the base claims 1 and 5 which patentably distinguish over Carpenter et al. and Irie et al. as discussed above.

Christie and Hooks et al. are discussed above, and are shown to fail to teach or suggest the subject matter of the base claims 1 and 5. Likewise as before, *prima facie* obviousness of the combination has not been shown.

Therefore, it is submitted that claims 3 and 7 patentably distinguish over Carpenter et al., Irie et al., Christie and Hooks et al., taken singly or in any proper combination.

ITEM 7: REJECTION OF CLAIM 9 UNDER 35 USC § 103(a) AS BEING UNPATENTABLE OVER CARPENTER ET AL. IN VIEW OF IRIE ET AL. AND PONG

Claim 9 is dependent upon the base claim 5 which patentably distinguishes over Carpenter at al. and Irie et al. as discussed above.

Pong merely proposes an apparatus for implementing a snoop protocol in a multiprocessor system without the use of snoop-in and snoop-out logic units. Each node of the multiprocessor system includes a memory access unit having an export cache that stores identifiers associated with data blocks that have been modified by another node. Each data block in a main memory unit of the node is associated with a state bit that indicates whether the data block is valid or invalid. The export cache and the state of each memory data block is used to determine whether a node should transmit a fetched data block to an initiator node in response to a read miss transaction, so as to reduce the bus traffic.

Pong does not teach or suggest the use of a preread and a buffer within the crossbar module to hold the data preread from a system module, other than the arbitrary system module, as recited in the base claim 5. Moreover, *prima facie* obviousness of the combination has not been shown.

Therefore, it is submitted that claim 9 patentably distinguishes over Carpenter at al., Irie et al. and Pong, taken singly or in any proper combination.

ITEM 8: REJECTION OF CLAIMS 10 AND 11 UNDER 35 USC § 103(a) AS BEING UNPATENTABLE OVER CARPENTER ET AL. IN VIEW OF IRIE ET AL. AND VENKITAKRISHNAN

Claim 10 is dependent upon the base claim 5 which patentably distinguishes over Carpenter et al. and Irie et al. as discussed above.

Venkitakrishnan merely proposes a crossbar switch which is implemented by a plurality of parallel chips which are programmable to couple to every node in the system, so as to provide a flexible structure that allows dynamic programming of the data routing and enable support of different network architectures.

Venkitakrishnan does not teach or suggest the use of a preread and a buffer within the crossbar module to hold the data preread from a system module other than the arbitrary system module, as recited in the base claim 5. As before, *prima facie* obviousness of the combination has not been shown.

Therefore, it is submitted that claim 10 patentably distinguishes over Carpenter et al., Irie et al.. and Venkitakrishnan, taken singly or in any proper combination.

With regard to claim 11, Carpenter at al., Irie et al. and Venkitakrishnan all fail to teach or suggest the use of a preread and a buffer within the crossbar module to hold the data preread from a system module other than the arbitrary system module.

NEW CLAIMS 12-17

New claims 12-17 recite further features of the present invention which are fully supported by the original disclosure on page 9, line 21 to page 11, line 35 of the specification, for example, and contain no new matter.

It is respectfully submitted that the new claims 12-17 are also allowable over the prior art of record.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: June 11, 2003

By: _____
H. J. Staas
Registration No. 22,010

700 Eleventh Street, NW, Suite 500
Washington, D.C. 20001
(202) 434-1500

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE CLAIMS:

Please AMEND the following claims:

1. (ONCE AMENDED) A memory access method for a multiprocessor system which includes a plurality of system modules coupled via a crossbar module, each of the system modules including a buffer which holds data and a plurality of processors having a cache memory which temporarily holds data, said memory access method comprising:

[a step,] responsive to a read request from a processor within an arbitrary system module, holding data preread from a system module, other than the arbitrary system module, in a buffer within the crossbar module.

2. (ONCE AMENDED) The memory access method as claimed in claim 1, further comprising:

[a step of] setting information indicating whether or not to carry out a data preread with respect to the arbitrary system module, depending on a program which is executed by one or a plurality of processors within the arbitrary system module.

3. (ONCE AMENDED) The memory access method as claimed in claim 2, further comprising:

[a step of] adding, to a data transfer of the preread data, a priority which is lower than a priority of a normal data transfer.

4. (ONCE AMENDED) The memory access method as claimed in claim 1, further comprising:

[a step of] adding, to a data transfer of the preread data, a priority which is lower than a priority of a normal data transfer.

5. (ONCE AMENDED) A multiprocessor system, comprising:

a plurality of system modules;

at least one crossbar module; and

a bus coupling the system modules and the crossbar module,

each of the system modules including a buffer which holds data, a plurality of

processors each having a cache memory which temporarily holds data, and a control unit which controls input and output of data with respect to the system module to which the control unit belongs,

a data transfer between two system modules being made via the crossbar module, and

said crossbar module including a buffer which holds data preread from a system module, other than an arbitrary system module, [in] responsive to a read request from a processor within the arbitrary system module.

6. (ONCE AMENDED) The multiprocessor system as claimed in claim 5, wherein the arbitrary system module includes [means for] a unit setting information indicating whether or not to carry out a data preread with respect to the arbitrary system module, depending on a program which is executed by one or a plurality of processors within the arbitrary system module.

7. (ONCE AMENDED) The multiprocessor system as claimed in claim 6, wherein each of the system modules further includes [means for] a unit adding, to a data transfer of the preread data, a priority which is lower than a priority of a normal data transfer.

8. (ONCE AMENDED) The multiprocessor system as claimed in claim 5, wherein each of the system modules further includes [means for] a unit adding, to a data transfer of the preread data, a priority which is lower than a priority of a normal data transfer.

9. (ONCE AMENDED) The multiprocessor system as claimed in claim 5, wherein one of the system modules, which has a memory with a requested address of the read request, includes [means for] a unit starting a data preread at a timing before detecting a state of the cache memory included therein.

10. (AS UNAMENDED) The multiprocessor system as claimed in claim 5, wherein:
the plurality of system modules, the crossbar module, and the bus form a node; and
a plurality of nodes are coupled via the crossbar module of adjacent nodes.

11. (ONCE AMENDED) A multiprocessor system, comprising:

a plurality of nodes each including a plurality of system modules, a crossbar module, and a bus coupling the system modules and the crossbar module within each node; and

a bus coupling adjacent nodes via the crossbar modules of the adjacent nodes,

each of the system modules including a buffer which holds data, a plurality of processors each having a cache memory which temporarily holds data, and a control unit which controls input and output of data with respect to the system module to which the control unit belongs,

a data transfer between two system modules being made via at least one crossbar module,

said crossbar module including a buffer which holds data preread from a system module, other than an arbitrary system module, in responsive to a read request from a processor within the arbitrary system module unit which controls input and output of data with respect to the system module to which the control unit belongs,

a data transfer between two system modules being made via at least one crossbar module, and

said crossbar module including a buffer which holds data preread from a system module, other than an arbitrary system module, [in] responsive to a read request from a processor within the arbitrary system module.

12. (NEW) The memory access method as claimed in claim 1, wherein said holding data further comprises:

generating the read request from the arbitrary system module if a mishit occurs in the arbitrary system module;

prereading the data requested by the read request in the system module, other than the arbitrary system module, and storing the requested data; and

transferring the preread data from the system module, other than the arbitrary system module, to the buffer within the crossbar module, with a priority lower than a priority of a normal data transfer between the system modules and the crossbar module.

13. (NEW) The memory access method as claimed in claim 12, further comprising:

transferring the preread data stored in the buffer within the crossbar module to the arbitrary system module with a priority lower than the priority of the normal data transfer

between the system modules and the crossbar module.

14. (NEW) The multiprocessor system as claimed in claim 5, wherein:

the arbitrary system module generates the read request if a mishit occurs in the arbitrary system module;

the system module, other than the arbitrary system module, storing the data requested by the read request prereads the requested data: and

the system module, other than the arbitrary system module, transfers the preread data to the buffer within the crossbar module, with a priority lower than a priority of a normal data transfer between the system modules and the crossbar module.

15. (NEW) The multiprocessor system as claimed in claim 14, wherein the crossbar module transfers the preread data stored in the buffer thereof to the arbitrary system module with a priority lower than the priority of the normal data transfer between the system modules and the crossbar module.

16. (NEW) The multiprocessor system as claimed in claim 11, wherein:

the arbitrary system module generates the read request if a mishit occurs in the arbitrary system module;

the system module, other than the arbitrary system module, storing the data requested by the read request prereads the requested data: and

the system module, other than the arbitrary system module, transfers the preread data to the buffer within the crossbar module, with a priority lower than a priority of a normal data transfer between the system modules and the crossbar module.

17. (NEW) The memory access method as claimed in claim 16, wherein the crossbar module transfers the preread data stored in the buffer thereof to the arbitrary system module with a priority lower than the priority of the normal data transfer between the system modules and the crossbar module.

18. (NEW) A memory access method for a multiprocessor system which includes a plurality of system modules coupled via a crossbar module, each of the system modules including a buffer which holds data and a plurality of processors having a cache memory which

temporarily holds data, said memory access method comprising:

responsive to a read request from a processor within an arbitrary system module, holding data preread from a system module, other than the arbitrary system module, in a buffer within the crossbar module.

19. (NEW) The memory access method as claimed in claim 18, further comprising:

setting information indicating whether or not to carry out a data preread with respect to the arbitrary system module, depending on a program which is executed by one or a plurality of processors within the arbitrary system module.

20. (NEW) The memory access method as claimed in claim 19, further comprising:

adding, to a data transfer of the preread data, a priority which is lower than a priority of a normal data transfer.

21. (NEW) The memory access method as claimed in claim 18, further comprising:

adding, to a data transfer of the preread data, a priority which is lower than a priority of a normal data transfer.

22. (NEW) A multiprocessor system, comprising:

a plurality of system modules;

at least one crossbar module; and

a bus coupling the system modules and the crossbar module,

each of the system modules including a buffer which holds data, a plurality of processors each having a cache memory which temporarily holds data, and a control unit which controls input and output of data with respect to the system module to which the control unit belongs,

a data transfer between two system modules being made via the crossbar module,

said crossbar module including a buffer which holds data preread from a system module, other than an arbitrary system module, in responsive to a read request from a processor within the arbitrary system module.

23. (NEW) The multiprocessor system as claimed in claim 22, wherein the arbitrary system module includes a unit setting information indicating whether or not to carry out a data preread with respect to the arbitrary system module, depending on a program which is executed by one or a plurality of processors within the arbitrary system module.

24. (NEW) The multiprocessor system as claimed in claim 23, wherein each of the system modules further includes a unit adding, to a data transfer of the preread data, a priority which is lower than a priority of a normal data transfer.

25. (NEW) The multiprocessor system as claimed in claim 22, wherein each of the system modules further includes a unit adding, to a data transfer of the preread data, a priority which is lower than a priority of a normal data transfer.

26. (NEW) The multiprocessor system as claimed in claim 22, wherein one of the system modules, which has a memory with a requested address of the read request, includes a unit starting a data preread at a timing before detecting a state of the cache memory included therein.

27. (NEW) The multiprocessor system as claimed in claim 22, wherein:
the plurality of system modules, the crossbar module, and the bus form a node; and
a plurality of nodes are coupled via the crossbar module of adjacent nodes.

28. (NEW) A multiprocessor system, comprising:
a plurality of nodes each including a plurality of system modules, a crossbar module, and a bus coupling the system modules and the crossbar module within each node; and
a bus coupling adjacent nodes via the crossbar modules of the adjacent nodes,
each of the system modules including a buffer which holds data, a plurality of processors each having a cache memory which temporarily holds data, and a control unit which controls input and output of data with respect to the system module to which the control unit belongs,

a data transfer between two system modules being made via at least one crossbar module,

said crossbar module including a buffer which holds data preread from a system module, other than an arbitrary system module, responsive to a read request from a processor within the arbitrary system module unit which controls input and output of data with respect to the system module to which the control unit belongs,

a data transfer between two system modules being made via at least one crossbar module, and

said crossbar module including a buffer which holds data preread from a system module, other than an arbitrary system module, in responsive to a read request from a processor within the arbitrary system module.

29. (NEW) The memory access method as claimed in claim 18, wherein said holding data includes:

generating the read request from the arbitrary system module if a mishit occurs in the arbitrary system module;

prereading the data requested by the read request in the system module, other than the arbitrary system module, and storing the requested data; and

transferring the preread data from the system module, other than the arbitrary system module, to the buffer within the crossbar module, with a priority lower than a priority of a normal data transfer between the system modules and the crossbar module.

30. (NEW) The memory access method as claimed in claim 29, further comprising:

transferring the preread data stored in the buffer within the crossbar module to the arbitrary system module with a priority lower than the priority of the normal data transfer between the system modules and the crossbar module.

31. (NEW) The multiprocessor system as claimed in claim 22, wherein:

the arbitrary system module generates the read request if a mishit occurs in the arbitrary system module;

the system module, other than the arbitrary system module, storing the data requested by the read request prereads the requested data: and

the system module, other than the arbitrary system module, transfers the preread data to the buffer within the crossbar module, with a priority lower than a priority of a normal data transfer between the system modules and the crossbar module.

32. (NEW) The multiprocessor system as claimed in claim 31, wherein the crossbar module transfers the preread data, stored in the buffer thereof, to the arbitrary system module with a priority lower than the priority of the normal data transfer between the system modules and the crossbar module.

33. (NEW) The multiprocessor system as claimed in claim 28, wherein:

the arbitrary system module generates the read request if a mishit occurs in the arbitrary system module;

the system module, other than the arbitrary system module, storing the data requested by the read request prereads the requested data: and

the system module, other than the arbitrary system module, transfers the preread data to the buffer within the crossbar module, with a priority lower than a priority of a normal data transfer between the system modules and the crossbar module.

34. (NEW) The memory access method as claimed in claim 33, wherein the crossbar module transfers the preread data stored in the buffer thereof to the arbitrary system module with a priority lower than the priority of the normal data transfer between the system modules and the crossbar module.